

WHAT IS CLAIMED IS:

1. A method for creating a proxy object capable of communication with an external entity, comprising:

specifying a proxy object definition for the proxy object wherein the proxy object definition defines a first function for communicating with the external entity;

specifying at least one implementation class for the proxy object definition, wherein the at least one implementation class does not implement the first function; and

wherein the at least one implementation class includes functionality to support one of: proxy object design, software compilation and software execution.

2. The method of claim 1, further comprising:

compiling the proxy object definition and/or the at least one implementation class.

3. The method of claim 1 wherein:

the at least one implementation class is specified with a source code annotation.

4. The method of claim 1, further comprising:

specifying at least one callback declaration.

5. The method of claim 1 wherein:

the functionality to proxy object software design includes a wizard that can guide the creation of the proxy object.

6. The method of claim 1 wherein:

the functionality to support software compilation includes at least one function to validate function calls and property settings in source code.

7. The method of claim 1 wherein:

the functionality to support software execution includes at least one function to acquire and release at least one resource and includes the ability to invoke a

function on the external entity.

8. The method of claim 1 wherein:
the external entity can be one of: a database, a legacy system, and a software application.
9. The method of claim 1 wherein:
the proxy object can inherit from a proxy object interface declaration at least one of: a function, a property and a callback.
10. The method of claim 1 wherein:
the proxy object definition includes a declaration of the first function.
11. A method for a proxy object to communicate with an external entity wherein the proxy object has an implementation class, said method comprising:
invoking a first function on the external entity via the proxy object, wherein the first function is not defined by the implementation class; and
dynamically determining the whether or not the first function invocation is proper based on metadata derived from a proxy object definition.
12. The method of claim 11 wherein:
the metadata includes at least one of the following: a proxy object interface declaration, a property setting, a callback declaration, and a function declaration.
13. The method of claim 11, further comprising:
invoking via the proxy object a callback function in an application based on receipt of an asynchronous event from the external entity.
14. The method of claim 11 wherein:
the implementation class is specified with a source code annotation.
15. The method of claim 11 wherein:
the implementation class includes at least one function to acquire and release at least one resource and includes the ability to invoke a function on the external

entity.

16. The method of claim 11 wherein:
the external entity can be one of: a database, a legacy system, and a software application.
17. The method of claim 11 wherein:
the proxy object can inherit from a proxy interface declaration at least one of:
a function, a property and a callback.
18. The method of claim 11 wherein:
the proxy object definition includes a declaration of the first function.
19. A system for creating a proxy object capable of communication with an external entity, comprising:
a proxy object definition for the proxy object wherein the proxy object definition defines a first function for communicating with the external entity;
at least one implementation class for the proxy object definition, wherein the at least one implementation class does not implement the first function;
a compiler capable of compiling the proxy object definition and the at least one implementation class; and
wherein the at least one implementation class includes functionality to support one of: proxy object design, software compilation and software execution.
20. The system of claim 19, further comprising:
the compiler is capable of generating metadata that can be used by the proxy object at runtime to invoke the first function.
21. The system of claim 19 wherein:
the at least one implementation class is specified with a source code annotation.
22. The system of claim 19, further comprising:
at least one callback declaration.

23. The system of claim 19 wherein:
the functionality to proxy object software design includes a wizard that can guide the creation of the proxy object.
24. The system of claim 19 wherein:
the functionality to support software compilation includes at least one function to validate function calls and property settings in source code.
25. The system of claim 19 wherein:
the functionality to support software execution includes at least one function to acquire and release at least one resource and includes the ability to invoke a function on the external entity.
26. The system of claim 19 wherein:
the external entity can be one of: a database, a legacy system, and a software application.
27. The system of claim 19 wherein:
the proxy object can inherit from a proxy object interface declaration at least one of: a function, a property and a callback.
28. The system of claim 19 wherein:
the proxy object definition includes a declaration of the first function.
29. A system for an application to communicate with an external entity, comprising:
an asynchronous event router capable of accepting asynchronous events from the external entity and forwarding them to one of: the application and a proxy object;
the application capable of invoking a function on the proxy object and capable of accepting asynchronous events from the asynchronous event router;
the proxy object capable of accepting asynchronous events from the asynchronous event router and invoking the function on the external entity;
wherein if the function is not defined by the proxy object, the proxy object is

capable of dynamically determining whether or not the function invocation is proper based on the metadata.

30. The system of claim 29 wherein:
the external entity can be one of: a database, a legacy system, and a software application.

31. The system of claim 29 wherein:
the proxy object can inherit from a proxy object interface declaration at least one of: a function, a property, and a callback.

32. A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:
invoke a first function on an external entity via a proxy object, wherein the first function is not defined by the implementation class; and
dynamically determine the whether or not the first function invocation is proper based on metadata derived from a proxy object definition.

33. The machine readable medium of claim 32 wherein:
the metadata includes at least one of the following: a proxy object interface declaration, a property setting, a callback declaration, and a function declaration.

34. The machine readable medium of claim 32, further comprising instructions that when executed cause the system to:
invoke via the proxy object a callback function in an application based on receipt of an asynchronous event from the external entity.

35. The machine readable medium of claim 32 wherein:
the implementation class is specified with a source code annotation.

36. The machine readable medium of claim 32 wherein:
the implementation class includes at least one function to acquire and release at least one resource and includes the ability to invoke a function on the external entity.

37. The machine readable medium of claim 32 wherein:
the external entity can be one of: a database, a legacy system, and a software application.
38. The machine readable medium of claim 32 wherein:
the proxy object can inherit from a proxy interface declaration at least one of:
a function, a property and a callback.
39. The machine readable medium of claim 32 wherein:
the proxy object definition includes a declaration of the first function.
40. A computer data signal embodied in a transmission medium, comprising:
a code segment including instructions to invoke a first function on an external entity via a proxy object, wherein the first function is not defined by the implementation class; and
a code segment including instructions to dynamically determine the whether or not the first function invocation is proper based on metadata derived from a proxy object definition.
41. The signal of claim 40 wherein:
the metadata includes at least one of the following: a proxy object interface declaration, a property setting, a callback declaration, and a function declaration.
42. The signal of claim 40, further comprising:
a code segment including instructions to invoke via the proxy object a callback function in an application based on receipt of an asynchronous event from the external entity.
43. The signal of claim 40 wherein:
the implementation class is specified with a source code annotation.
44. The signal of claim 40 wherein:
the implementation class includes at least one function to acquire and release

at least one resource and includes the ability to invoke a function on the external entity.

45. The signal of claim 40 wherein:
the external entity can be one of: a database, a legacy system, and a software application.

46. The signal of claim 40 wherein:
the proxy object can inherit from a proxy interface declaration at least one of: a function, a property and a callback.

47. The signal of claim 40 wherein:
the proxy object definition includes a declaration of the first function.